Case Study - Recommendations for Marketing Strategy

Naska Kvarantan

2023-11-27

The mission statement

Bellabeat is a high tech manufacturer of health-focused products for women, which include a Bellabeat App, smart watch, smart wellness tracker and a smart water bottle. These products provide their users with data related to their activity, sleep, stress, menstrual cycle and mindfulness habits. They want to analyse fitness data of non-Bellabeat smart device users in order to gain insight into how consumers are using their smart devices. Using this information the company would like to recieve high – level recomendations for Bellabeat's marketing strategy.

Key questions for the analysis:

1. What are some trends in smart device usage?
2. How could these trends apply to Bellabeat customers?
3. How could these trends help influence Bellabeat marketing strategy?

Ask

Business task

The business task at hand is to identify trends in non-Bellabeat smart device usage to identify trends and insights that can be applied to Bellabeat's products. Finally, these findings will be used to formulate recommendations for Bellabeat's marketing strategy.

Stakeholders

- Urška Sršen, Bellabeat's cofounder and Chief Creative Officer
- Sando Mur, Bellabeat's cofounder and key member of the Bellabeat executive team
- Bellabeat marketing analytics team

Prepare

Data Source

The dataset used for the analysis was retrieved from FitBit Fitness Tracker Data (CC0: Public Domain)

The dataset contains personal fitness tracking data from thirty Fitbit users. The data includes personal tracker-data, including minute-level output for physical activity, heart rate, and sleep monitoring. Furthermore, it includes information about daily activity, steps, and heart rate. The data was collected from 12.4.2016 to 12.5.2016.

The dataset is comprised of 18 .csv files.

Limitations of the data

Upon first review of the data, there are some limitations that are important to consider:

1. FitBit data does not include information about gender and age. Since Bellabeat is targeting mainly women with their products, this is an important factor to keep in mind when making recommendations based on this data.

2. The dataset may exhibit some bias since all participants were volunteers, they were not randomly selected from a wide population of FitBit users. Thus, the entire population of FitBit users might not be reflected in this sample.

3. The data was collected in 2016, which means it is not very current. The trends from 2016 might not be applicable to 2023.

Process

Data Cleaning Process

Data cleaning was done in RStudio. I started by installing a few useful packages.

```
# Set a CRAN mirror
options(repos = c(CRAN = "https://cloud.r-project.org"))

# Install and load the required packages

install.packages("tidyverse")
library(tidyverse)
install.packages("lubridate")
library(lubridate)
install.packages("dplyr")
library(dplyr)
```

Then I imported the data to analyse. For this case study, I decided to use a subset of the 18 files available.

```
DailyActivity <- read.csv("/Users/oblj-nkvarantan/Documents/CASE STUDY /Kaggle Case Study Data_Untouched
Heartrate <- read.csv("/Users/oblj-nkvarantan/Documents/CASE STUDY /Kaggle Case Study Data_Untouched/Fi
HourlyCalories <- read.csv("/Users/oblj-nkvarantan/Documents/CASE STUDY /Kaggle Case Study Data_Untouche
HourlyIntensities <- read.csv("/Users/oblj-nkvarantan/Documents/CASE STUDY /Kaggle Case Study Data_Untou
MinuteSleep <- read.csv("/Users/oblj-nkvarantan/Documents/CASE STUDY /Kaggle Case Study Data_Untouched/
DailySleep <- read.csv("/Users/oblj-nkvarantan/Documents/CASE STUDY /Kaggle Case Study Data_Untouched/Fi
```

Then, I used the str() function to check if all the variables were in the correct datatype. I noticed that the date column included both the date and time variables, which I didn't like. I split the column into two separate columns: Date and Time. While doing this I also changes the time format to 24h in order to remove AM/PM, and the dates from mm/dd/yyyy to yyyy/mm/dd format. Since the date columns were not the correct datatype - they were strings, I had to convert the strings to dates.

Example of process:

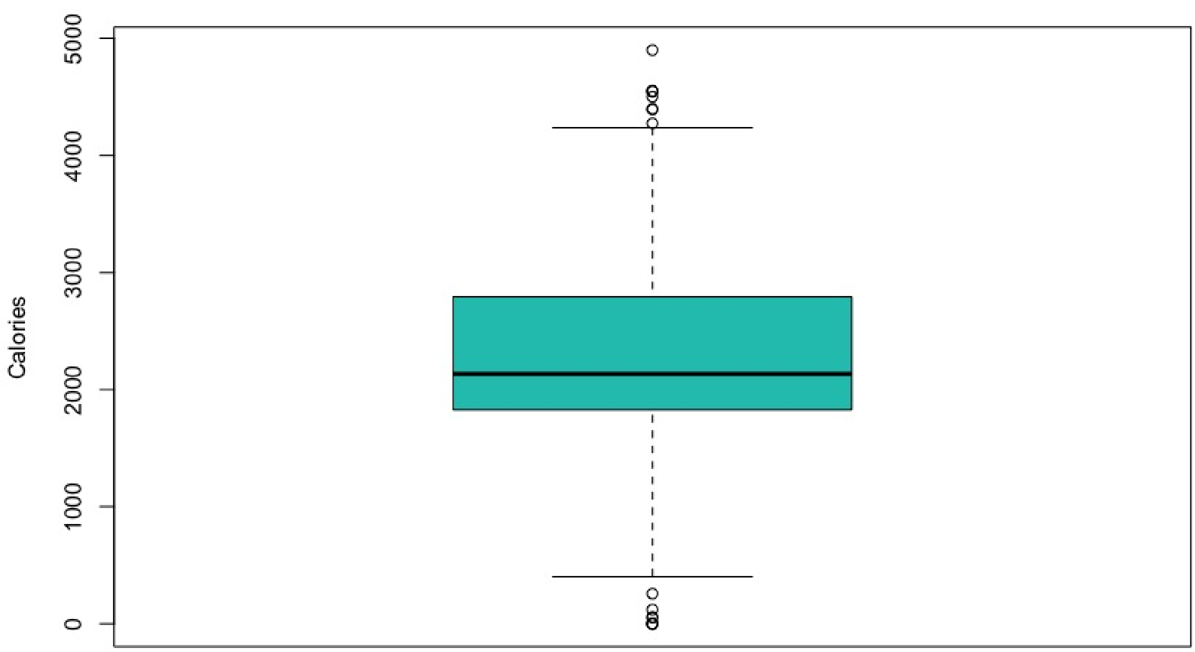
```
HeartRate <- read.csv(".../Documents/CASE STUDY /Kaggle Case Study Data_Untouched/Fitabase Data 4.12.16

HeartRate$Time <- as.POSIXct(HeartRate$Time, format = "%m/%d/%Y %I:%M:%S %p")
```

```r
HeartRate <- HeartRate %>%
  mutate(
    Date = as.Date(Time),
    Time = format(Time, format = "%H:%M:%S")
  ) %>%
  select(Id, Date, Time, Value)

write.csv(HeartRate, "/.../Documents/CASE STUDY /Kaggle Case Study Data_Untouched/Fitabase Data 4.12.16
```

Secondly, I checked for missing values.

Thirdly, I used the summary() function to check the max/min points, means and median for each relevant variable, to see if there are big deviations. I noticed that in the DailyActivity data set, there were some values that did not make any sense. The Calories column shows the amount of calories burned in a day for each participant, including Basal Metabolic Rate (BMR). Therefore, I decided to drop all entries that had a value of 0, since it is impossible that the BMR alone would amount to 0. Next, I noticed there are more values which were very unlikely, so I decided to check for outliers in the Calories column with a boxplot:

```r
boxplot(DailyActivity$Calories,
        ylab = "Calories", col="#20C4BC"
)

boxplot.stats(DailyActivity$Calories)$out
```



Based on the boxplot and based on it being very unlikely that someone would burn less than 500 kcal in one day (due to the included BMR measurements), I decided to drop all values below 500 kcal from the DailyActivity dataset, because it means either the measurements weren't collected for the whole day or something went wrong.In both cases, the values for other variables are probably also not going to be accurate.

With my data cleaned and prepared I moved to the analysis phase.

Analyse

Let's first do a summary of the data that I'll be using for the analysis.

```
summary(DailyActivity)
```

```
##        Id             ActivityDate         TotalSteps    TotalDistance
##  Min.   :1.504e+09   Length:940         Min.   :    0   Min.   : 0.000
##  1st Qu.:2.320e+09   Class :character   1st Qu.: 3790   1st Qu.: 2.620
##  Median :4.445e+09   Mode  :character   Median : 7406   Median : 5.245
##  Mean   :4.855e+09                      Mean   : 7638   Mean   : 5.490
##  3rd Qu.:6.962e+09                      3rd Qu.:10727   3rd Qu.: 7.713
##  Max.   :8.878e+09                      Max.   :36019   Max.   :28.030
##  TrackerDistance  LoggedActivitiesDistance VeryActiveDistance
##  Min.   : 0.000   Min.   :0.0000           Min.   : 0.000
##  1st Qu.: 2.620   1st Qu.:0.0000           1st Qu.: 0.000
##  Median : 5.245   Median :0.0000           Median : 0.210
##  Mean   : 5.475   Mean   :0.1082           Mean   : 1.503
##  3rd Qu.: 7.710   3rd Qu.:0.0000           3rd Qu.: 2.053
##  Max.   :28.030   Max.   :4.9421           Max.   :21.920
##  ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
##  Min.   :0.0000           Min.   : 0.000      Min.   :0.000000
##  1st Qu.:0.0000           1st Qu.: 1.945      1st Qu.:0.000000
##  Median :0.2400           Median : 3.365      Median :0.000000
##  Mean   :0.5675           Mean   : 3.341      Mean   :0.001606
##  3rd Qu.:0.8000           3rd Qu.: 4.782      3rd Qu.:0.000000
##  Max.   :6.4800           Max.   :10.710      Max.   :0.110000
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0        Min.   :   0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0        1st Qu.: 729.8
##  Median :  4.00    Median :  6.00      Median :199.0        Median :1057.5
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8        Mean   : 991.2
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:1229.5
##  Max.   :210.00    Max.   :143.00      Max.   :518.0        Max.   :1440.0
##     Calories
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

- From a first glance at the data, we can see that the average day of steps per day is 7698.
- On average 21.38 minutes in a day were spent doing very active activity, 13.56 minutes of fairly active activity and 192.8 minutes of lightly active minutes. Most of the active time was spent doing light activity.
- Participants were sedentary for the majority of their day (991.2 min = 16.52 h).

```r
summary(Heartrate)
```

```
##        Id                  Date                Time                Value
##  Min.   :2.022e+09   Length:2483658     Length:2483658     Min.   : 36.00
##  1st Qu.:4.388e+09   Class :character   Class :character   1st Qu.: 63.00
##  Median :5.554e+09   Mode  :character   Mode  :character   Median : 73.00
##  Mean   :5.514e+09                                         Mean   : 77.33
##  3rd Qu.:6.962e+09                                         3rd Qu.: 88.00
##  Max.   :8.878e+09                                         Max.   :203.00
```

The average heart rate of all participants in one month was 77.33.

FitBit measures sleep which is divided into 3 sleep states: asleep (1), restless (2) and awake (3). In order to find out the frequency of each sleep states, we have to count how many times each appears.

```r
table(MinuteSleep$value)
```

```
##
##      1      2      3
## 172480  14023   2018
```

The participants spend 91.49% of their sleep sleeping, 7.44% of their sleep was restless, and 1.07% was spent awake.

```r
summary(DailySleep)
```

```
##        Id                  Date                Time           TotalSleepRecords
##  Min.   :1.504e+09   Length:413         Length:413         Min.   :1.000
##  1st Qu.:3.977e+09   Class :character   Class :character   1st Qu.:1.000
##  Median :4.703e+09   Mode  :character   Mode  :character   Median :1.000
##  Mean   :5.001e+09                                         Mean   :1.119
##  3rd Qu.:6.962e+09                                         3rd Qu.:1.000
##  Max.   :8.792e+09                                         Max.   :3.000
##  TotalMinutesAsleep TotalTimeInBed
##  Min.   : 58.0      Min.   : 61.0
##  1st Qu.:361.0      1st Qu.:403.0
##  Median :433.0      Median :463.0
##  Mean   :419.5      Mean   :458.6
##  3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :796.0      Max.   :961.0
```

On average the participants slept approx. 7 hours a day.

```r
summary(HourlyIntensities)
```

```
##        Id                  Date                Time           TotalIntensity
##  Min.   :1.504e+09   Length:22099       Length:22099       Min.   :  0.00
##  1st Qu.:2.320e+09   Class :character   Class :character   1st Qu.:  0.00
##  Median :4.445e+09   Mode  :character   Mode  :character   Median :  3.00
##  Mean   :4.848e+09                                         Mean   : 12.04
##  3rd Qu.:6.962e+09                                         3rd Qu.: 16.00
```

```
##  Max.   :8.878e+09                                      Max.   :180.00
##  AverageIntensity
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0500
##  Mean   :0.2006
##  3rd Qu.:0.2667
##  Max.   :3.0000
```

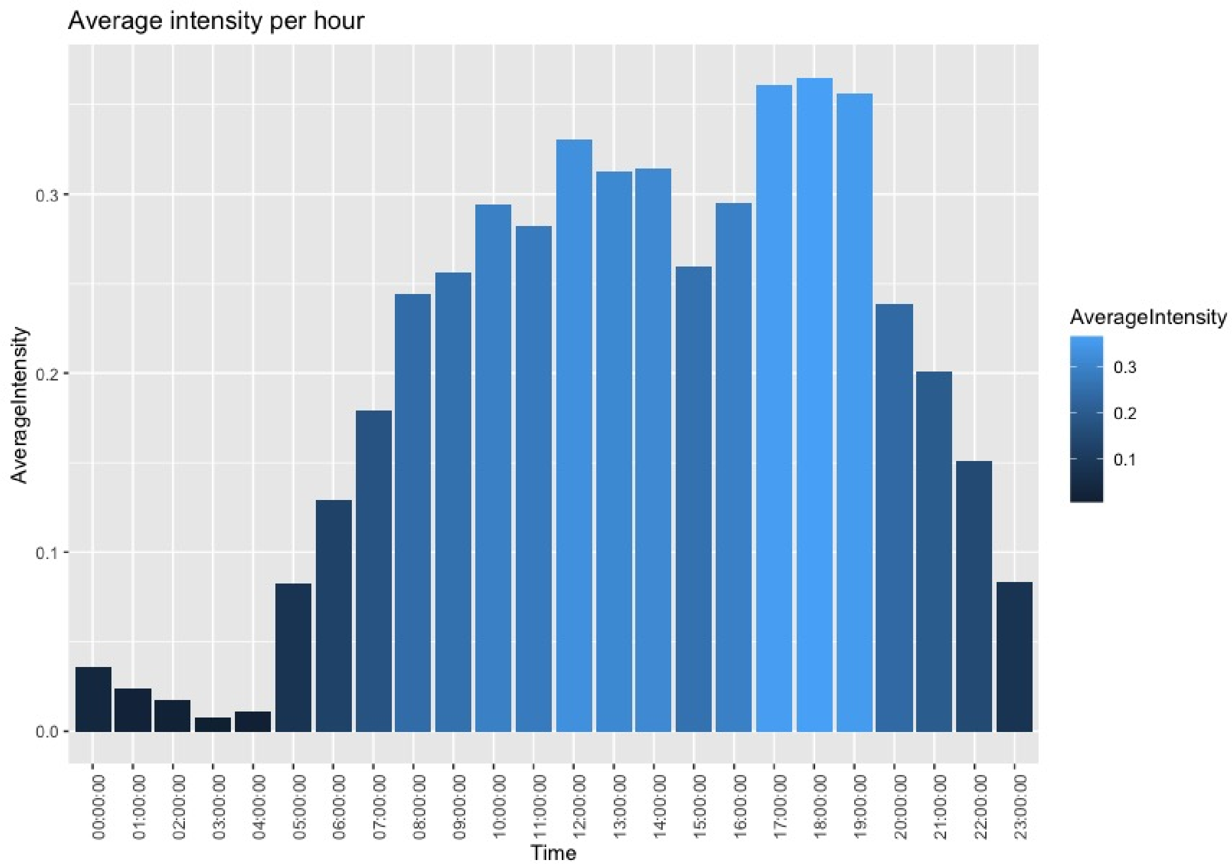
After a quick look at the data, there are a few questions I want to explore:

1. During which part of the day and which day of the week do people tend to exercise the most?
2. How does the level of activity correlate to calories burned?
3. Does the level of activity in a day influence sleep?
4. Does average heart rate compare to activity level?

I will first focus on some aspects of the data to see if there are any trends that I can identify. It might be interesting to focus more on physical activity, sleep and heart rate.

Data about physical activity (intenisity, steps)

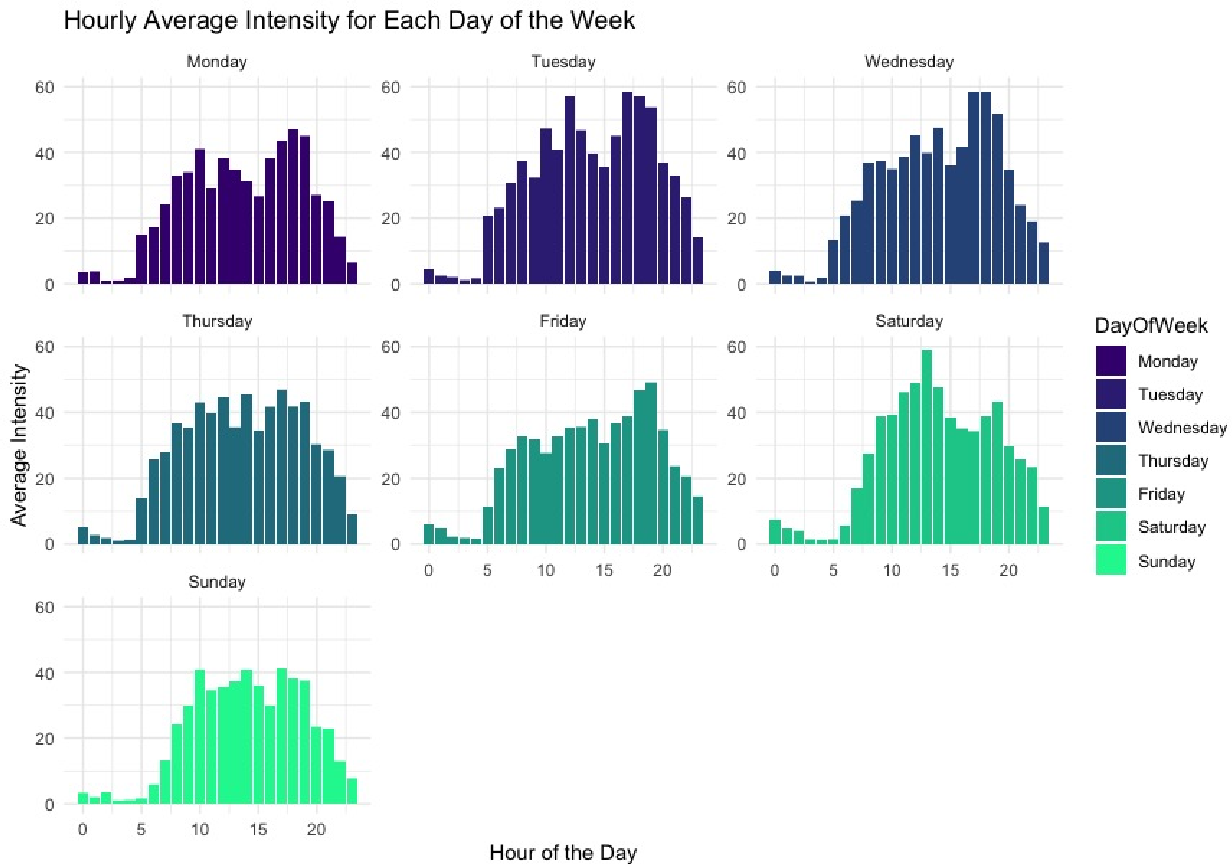
```
HourlyIntensities %>%
  select(TotalIntensity, AverageIntensity) %>%
  summary()

Average_Intensity_Hour <- HourlyIntensities %>%
  group_by(Time) %>%
  summarise(AverageIntensity = mean(AverageIntensity))

ggplot(data=Average_Intensity_Hour) +
  geom_col(mapping = aes(x=Time, y=AverageIntensity, fill=AverageIntensity)) +
  theme(axis.text.x = element_text(angle = 90))+
  labs(title = "Average intensity per hour")
```



From the graph we see a noticeable spike in average physical activity intensity between 17:00 and 19:00. Which makes sense since a lot of people work shifts from 09:00 to 17:00 and after that they might go to their preferred methods of physical activity.

But let's have a look if the average intensities differed between days in the week.

```
HourlyIntensities$DayOfWeek <- factor(HourlyIntensities$DayOfWeek, levels = c("Monday", "Tuesday", "Wedn

plots <- HourlyIntensities %>%
  ggplot(aes(x = hour(Time), y = AverageIntensity, group = DayOfWeek, fill=DayOfWeek)) +
  geom_col() +
  scale_fill_manual(values = week_day_colors_7) +
  facet_wrap(~ DayOfWeek, scales = 'free_y') +
  labs(title = "Hourly Average Intensity for Each Day of the Week",
       x = "Hour of the Day",
       y = "Average Intensity") +
  theme_minimal()

print(plots)
```

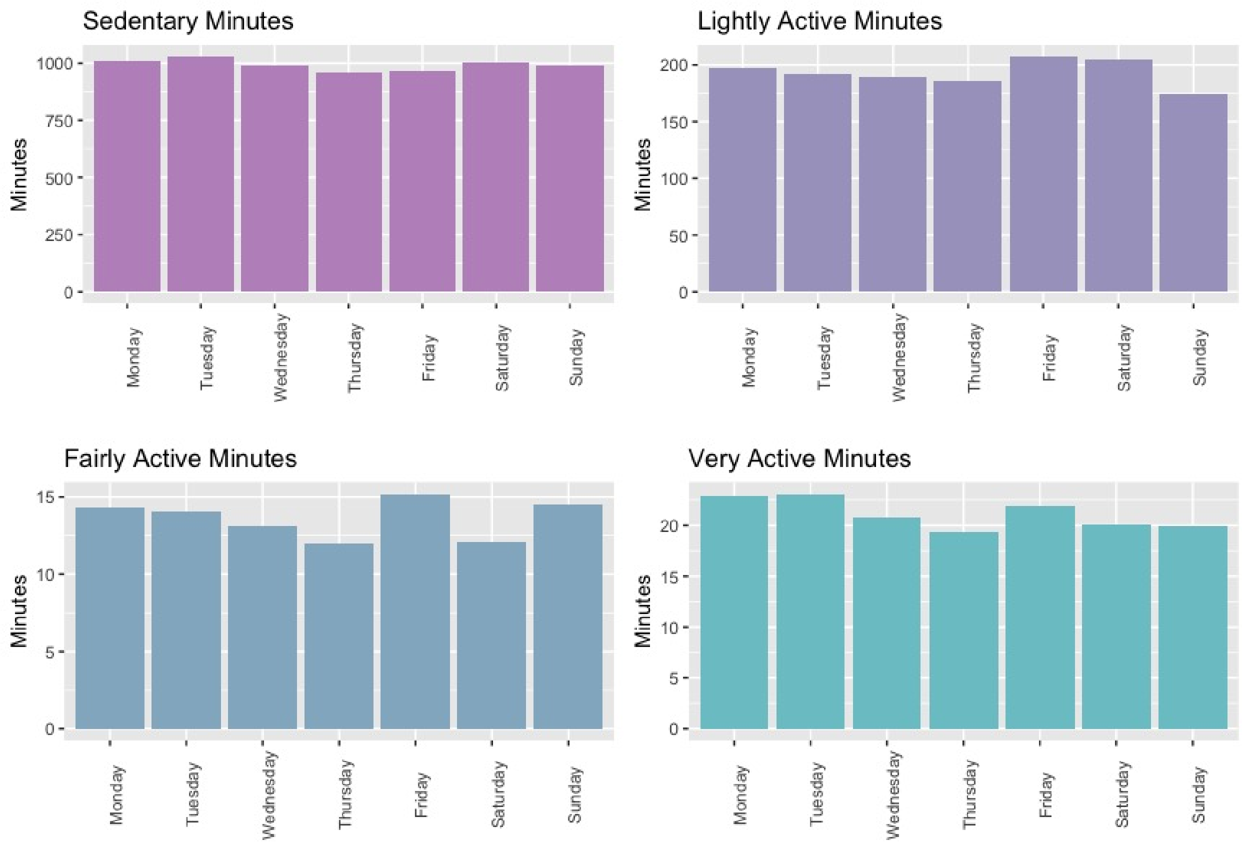


From the graphs above it is clear that the distribution of hourly intensities is not the same for week days and weekend days. It is interesting to note that graphs for Monday, Tuesday, Wednesday and Friday look quite similar, with the highest average intensity being observed from 17:00 to 19:00. However, the average intensity recorded was highest on Tuesday, Wednesday and Saturday.

Next, I'll look at the difference in the level of physical activity throughout the week.

```
DailyIntensities <- mutate(DailyIntensities, DayOfWeek = weekdays(Date))

weekly_summary_intensities <- DailyIntensities %>%
  group_by(DayOfWeek) %>%
  summarise(AvgSedentaryMinutes = mean(SedentaryMinutes), AvgLightlyActiveMinutes = mean(LightlyActiveMi

weekly_summary_intensities$DayOfWeek <- factor(weekly_summary$DayOfWeek, levels = c("Monday", "Tuesday"

plot1 <- ggplot(weekly_summary_intensities, aes(x = DayOfWeek, y = AvgSedentaryMinutes, fill = AvgSeden
  geom_col(fill="#BE93C5") +
  theme(axis.text.x = element_text(angle = 90))+
  labs(title = "Sedentary Minutes", x="", y="Minutes")

plot2 <- ggplot(weekly_summary_intensities, aes(x = DayOfWeek, y = AvgLightlyActiveMinutes, fill = AvgLi
  geom_col(fill = "#A7A3C7") +
  theme(axis.text.x = element_text(angle = 90))+
  labs(title = "Lightly Active Minutes", x="", y="Minutes")

plot3 <- ggplot(weekly_summary_intensities, aes(x = DayOfWeek, y = AvgFairlyActiveMinutes, fill = AvgFa
```

```
  geom_col(fill="#91B4C9") +
  theme(axis.text.x = element_text(angle = 90))+
  labs(title = "Fairly Active Minutes",  x="", y="Minutes")

plot4 <- ggplot(weekly_summary_intensities, aes(x = DayOfWeek, y = AvgVeryActiveMinutes, fill = AvgVery
  geom_col(fill="#7BC6CC") +
  theme(axis.text.x = element_text(angle = 90))+
  labs(title = "Very Active Minutes",  x="", y="Minutes")

# Arrange plots side by side
grid.arrange(plot1, plot2, plot3, plot4, ncol = 2)
```

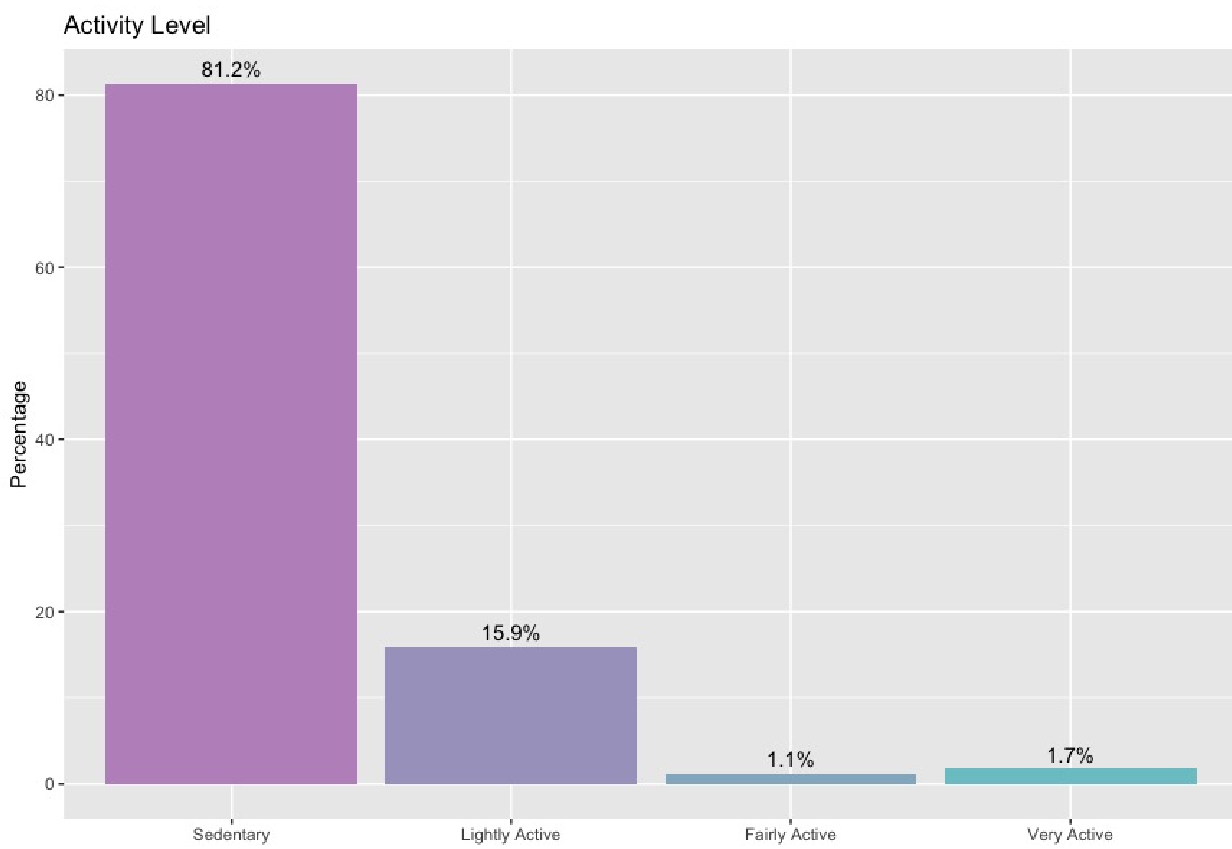


We can observe that sedentary minutes do not vary much throughout the week. But we do see that our participants were slightly less sedentary on average on Thursdays and Fridays. Furthermore, our participants spent more minutes being very active on Mondays and Tuesdays.

Next, I wanted to see the mean daily activity level of our participants.

```
sum_sedentary <- sum(DailyActivity$SedentaryMinutes)
sum_lightly <- sum(DailyActivity$LightlyActiveMinutes)
sum_fairly <- sum(DailyActivity$FairlyActiveMinutes)
sum_very <- sum(DailyActivity$VeryActiveMinutes)
total <- sum(DailyActivity$SedentaryMinutes)+sum(DailyActivity$LightlyActiveMinutes)+sum(DailyActivity$

col1 <- c("Sedentary", "Lightly Active", "Fairly Active", "Very Active")
col2 <- c(sum_sedentary/total*100, sum_lightly/total*100, sum_fairly/total*100, sum_very/total*100)
```

```
ActivityLevel <- data.frame(Activity_Level = col1, Percentage = col2)
ActivityLevel$Activity_Level <- factor(ActivityLevel$Activity_Level, levels = c("Sedentary", "Lightly Ac

ggplot(data=ActivityLevel, aes(x= Activity_Level, y = Percentage, fill = Activity_Level))+
  geom_col()+
  geom_text(aes(label =sprintf("%.1f%%", Percentage)), vjust = -0.5, color = "black")+
  scale_fill_manual(values = user_type_colors) +
  theme(legend.position = "none")+
  labs(title= "Activity Level", x = "", y= "Percentage")
```

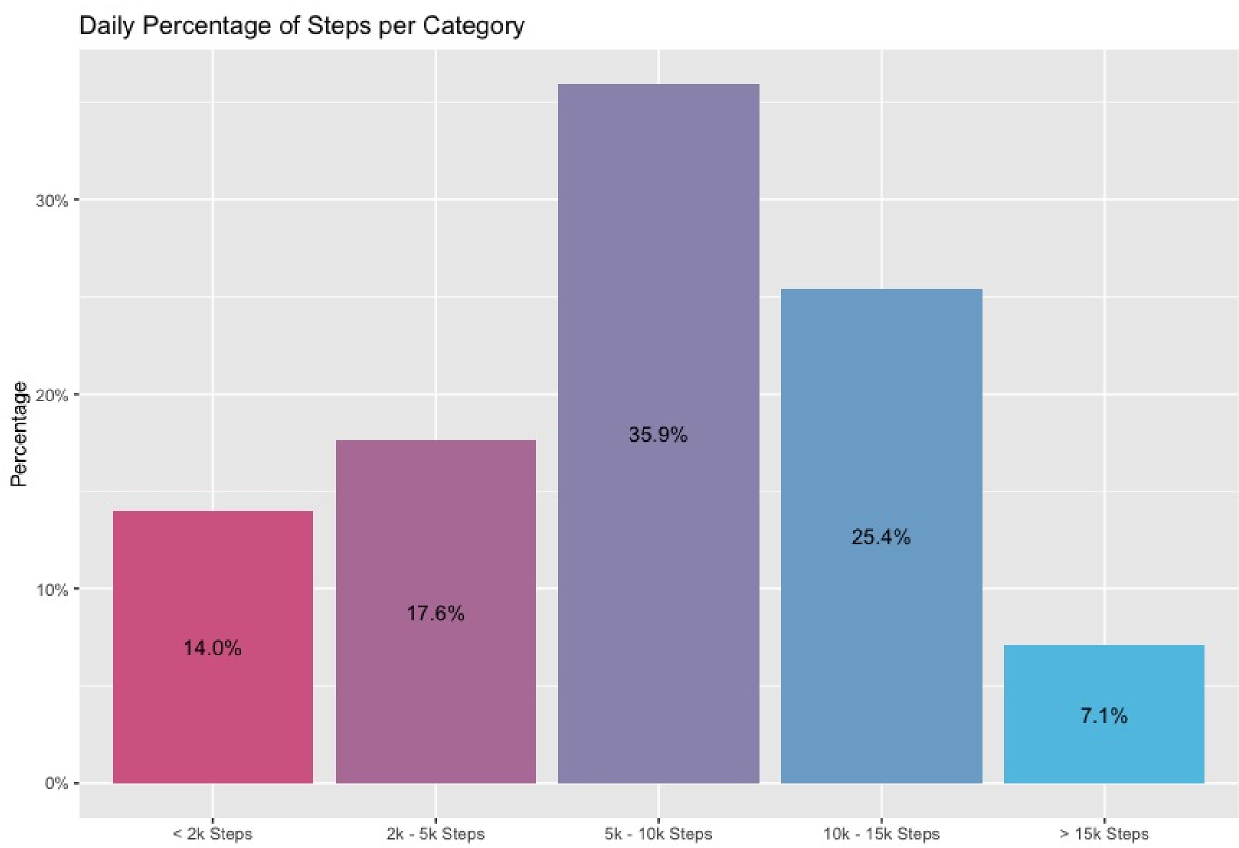


According to their data, it seems that on average people were sedentary for 81.2 % of their day, 15.9 % of the day being lightly active, 1.1 % of the day being fairly active and 1.7 % of the day being very active.

Next, I'll be looking into the daily step count. From the summary of the DailyActivity data set above I saw that the average daily step count was 7698. In order to have a better understanding of the distribution of the daily steps of our participants I decided to first divide them into 5 categories: < 2000 steps, between 2000 and 5000 steps, between 5000 and 10.000 steps, between 10.000 and 15.000 steps and more than 15.000 steps.

```
StepCategory_5 <- DailyActivity %>%
  mutate(
    Step_Category_5 = case_when(
      TotalSteps < 2000 ~ "< 2k Steps",
      TotalSteps >= 2000 & TotalSteps < 5000 ~ "2k - 5k Steps",
      TotalSteps >= 5000 & TotalSteps < 10000 ~ "5k - 10k Steps",
```

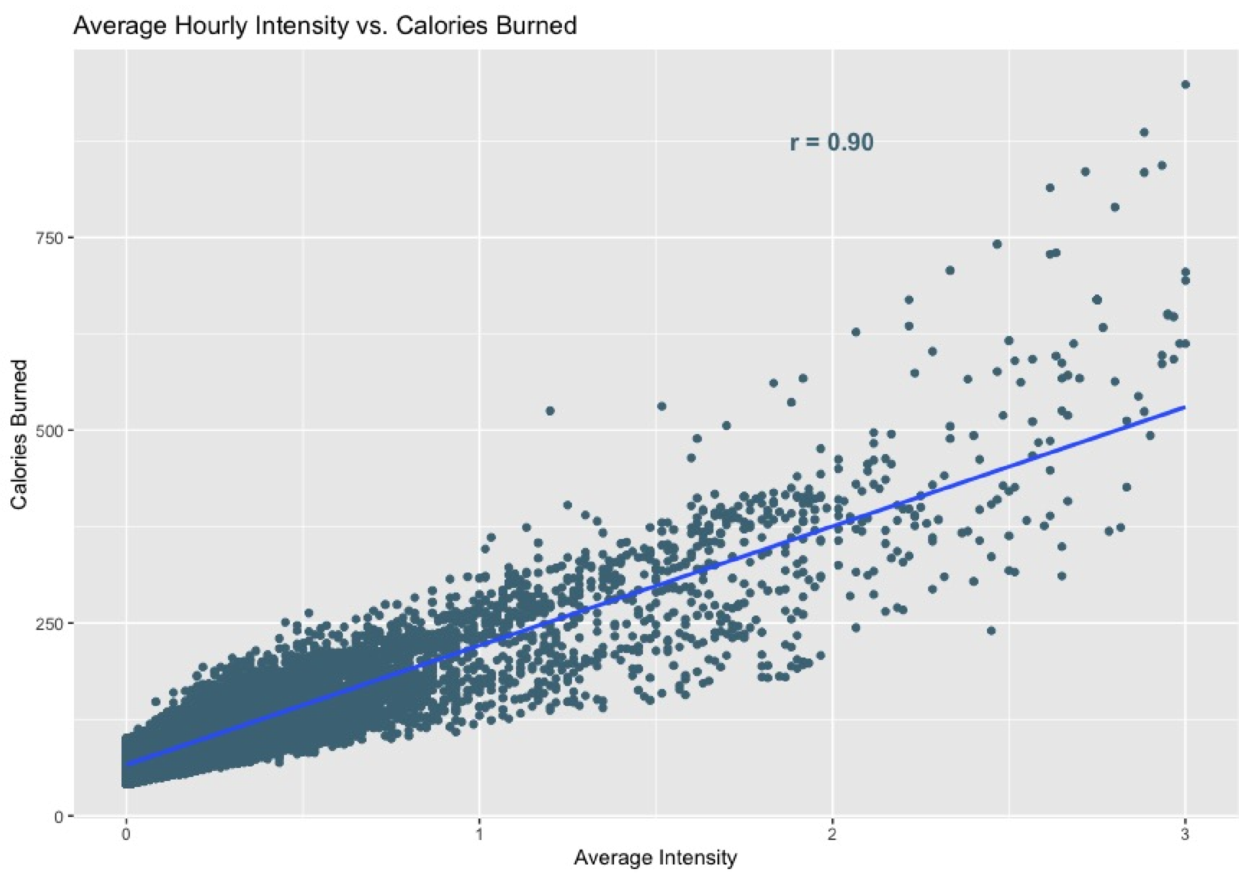
```r
      TotalSteps >= 10000 & TotalSteps < 15000 ~ "10k - 15k Steps",
      TotalSteps >= 15000 ~ "> 15k Steps")
  ) %>%
  select(Id, Step_Category_5) %>%
  count(Step_Category_5) %>%
  group_by(Step_Category_5) %>%
  arrange(desc(n))
print(StepCategory_5)

StepCategory_5$Step_Category_5 <- factor(StepCategory_5$Step_Category_5, levels = c("< 2k Steps", "2k -

colors_step_cat_5 <- c("< 2k Steps" = "#D66891", "2k - 5k Steps" = "#B87EA6", "5k - 10k Steps" = "#9A95

StepCategory_5 %>%
  group_by(Step_Category_5) %>%
  summarise(total = n) %>%
  mutate(totals = sum(total)) %>%
  group_by(Step_Category_5) %>%
  summarise(total_percent = total/totals) %>%
  ggplot(aes(Step_Category_5, y=total_percent, fill=Step_Category_5))+
  geom_col(show.legend = FALSE)+
  geom_text(aes(label = scales :: percent(total_percent)), position = position_stack(vjust = 0.5), colo
  scale_fill_manual(values = colors_step_cat_5)+
  scale_y_continuous(labels = scales::percent) +
  labs(title="Daily Percentage of Steps per Category", x="", y= "Percentage")
```



From the graph above it's pretty clear that the majority of our participants took between 5000 and 10.000 steps per day. Interestingly, the second most frequent number of steps was between 10.000 and 15.000 steps per day. Research on this topic suggests that healthy adults (aged 20 - 50 years) can take anywhere between 4000 and 18.000 steps/day, and that 10.000 steps/day is a reasonable target (Source: Locke et al., 2011.

How does the level of activity correlate to calories burned? Here I decided to look into the relationship between activity intensity and calories burned.

```
HourlyIntensitiesxHourlyCalories <- merge(HourlyIntensities, HourlyCalories, by=c('Id', 'Date', 'Time')
head(HourlyIntensitiesxHourlyCalories)

ggplot(data=HourlyIntensitiesxHourlyCalories, aes(x= AverageIntensity, y = Calories))+
  geom_point(color="#4a7384")+
  geom_smooth(method = "lm")+
  labs(title= "Average Hourly Intensity vs. Calories Burned", x = "Average Intensity", y= "Calories Bur
  annotate("text", x=2,y=875,label="r = 0.90", color="#4a7384", fontface="bold", size=4.5)
```

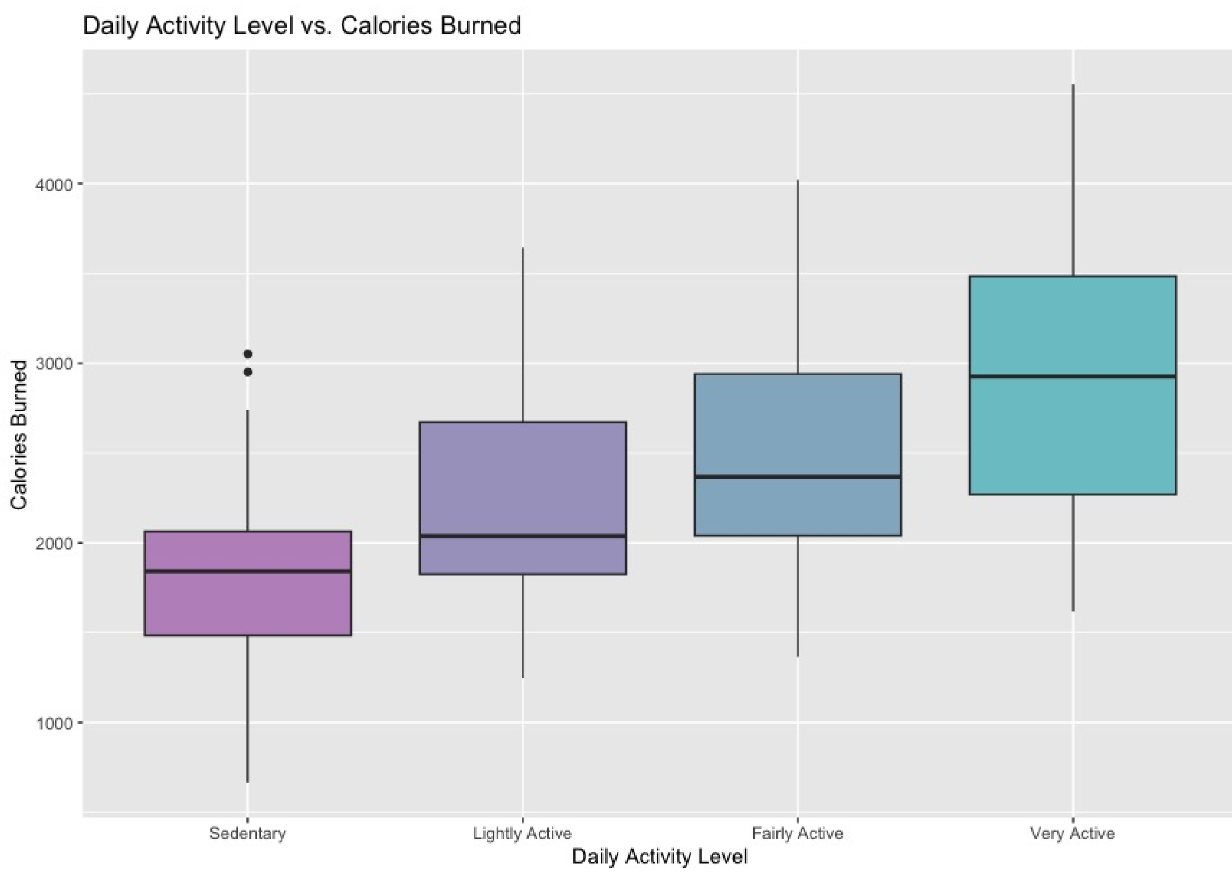


In the scatterplot above, a positive correlation can be observed between the amount of calories burned and the intensity of physical activity, which is to be expected. The higher the intensity of the activity, the more calories were burned.

Can a similar trend be observed with data for daily step count? Based on the study quoted above I decided to form 4 activity levels, which would match those of the FitBit measurements: Sedentary, Lightly Active, Fairly Active and Very Active. I decided that anything under 4000 steps/day would be labelled as

Sedentary; between 4000 and 8000 steps/day would be Lightly Active; between 8000 and 12000 steps/day would be Fairly Active and everything above 12000 steps/day would be Very Active. Next, I compared the level of activity with the calories burned.

```
DailyActivity2 <- DailyActivity %>%
  mutate(
  DailyActivityLevel = case_when(
    TotalSteps < 4000 ~ "Sedentary",
    TotalSteps >= 4000 & TotalSteps < 8000 ~ "Lightly Active",
    TotalSteps >= 8000 & TotalSteps < 12000 ~ "Fairly Active",
    TotalSteps >= 12000 ~ "Very Active"
    )
  )

DailyActivity2$DailyActivityLevel <- factor(DailyActivity2$DailyActivityLevel, levels = c("Sedentary",

ggplot(data=DailyActivity2, aes(x = DailyActivityLevel, y = Calories, fill = DailyActivityLevel)) +
  geom_boxplot() +
  scale_fill_manual(values = daily_activity_colors) +
  theme(legend.position = "none") +
  labs(title = "Daily Activity Level vs. Calories Burned", x = "Daily Activity Level", y= "Calories Bur

daily_activity_colors <- c("Sedentary" = "#BE93C5", "Lightly Active" = "#A7A3C7", "Fairly Active" = "#9
```



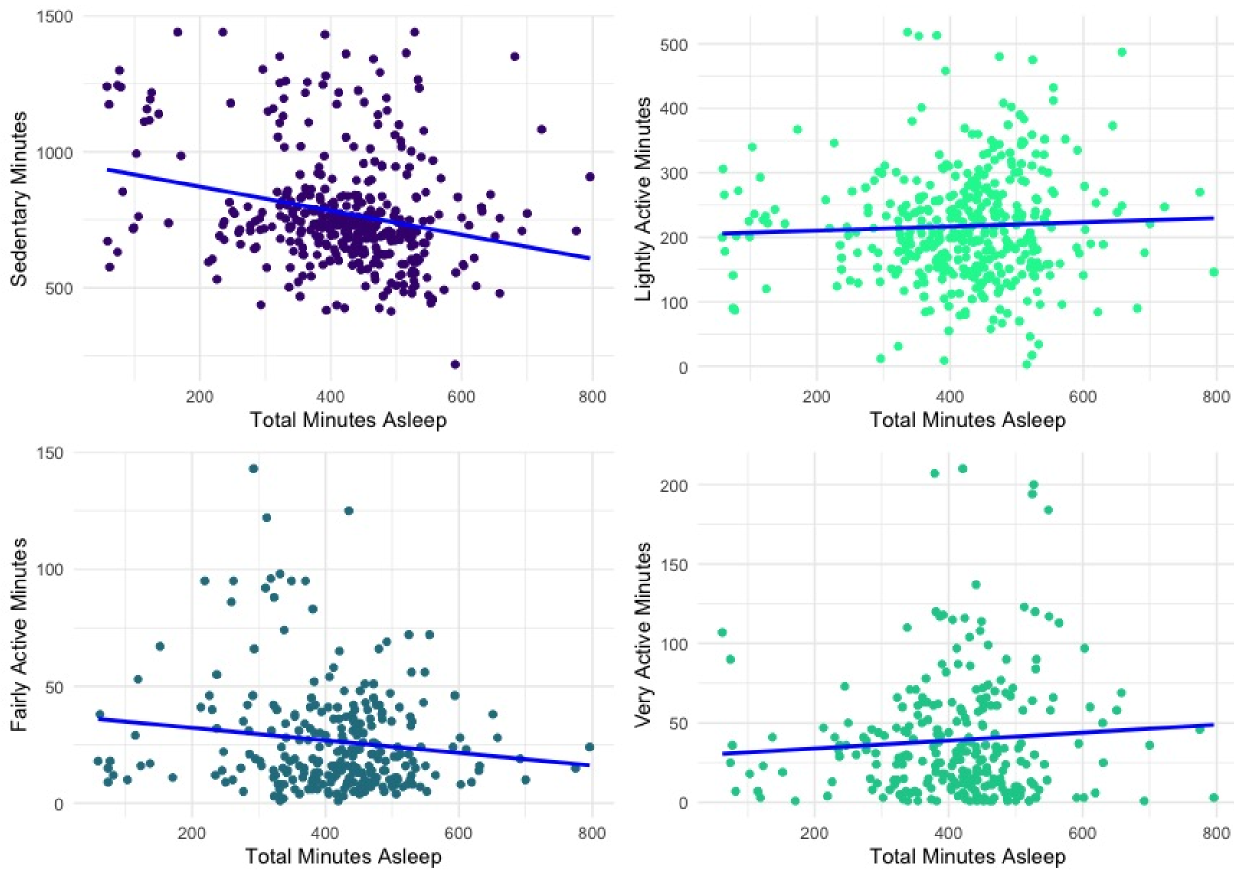
It is clear that the number of calories burned per day increases with the level of activity, which is defined by the number of steps taken per day.

Data about sleep

How does the level of activity affect sleep? To look into this, I first had to merge two datasets together: DailyActivity and DailySleep. Next, I had to remove the 0 values in the lightly, fairly and very active minutes columns.

```
ActivityxSleep <- merge(DailyActivity, DailySleep, by=c('Id', 'Date'))
head(ActivityxSleep)

ActivityxSleep_NZV_lightly <- ActivityxSleep %>% # NZV = no zero values for the lightly active minutes
  filter(LightlyActiveMinutes != 0)

ActivityxSleep_NZV_fairly <- ActivityxSleep %>% # NZV = no zero values for the fairly active minutes
  filter(FairlyActiveMinutes != 0)

ActivityxSleep_NZV_very <- ActivityxSleep %>% # NZV = no zero values for the very active minutes
  filter(VeryActiveMinutes != 0)

Sedentary_plot <- ggplot(ActivityxSleep, aes(x = TotalMinutesAsleep, y = SedentaryMinutes)) +
  geom_point(color ="#42047E") +
  scale_fill_manual(values = "#42047E") +
  geom_smooth(method = "lm", se=FALSE, color = "blue") +
  labs(
    x = "Total Minutes Asleep",
    y = "Sedentary Minutes"
  ) +
  theme_minimal()

Lightly_active_plot <- ggplot(ActivityxSleep_NZV_lightly, aes(x = TotalMinutesAsleep, y = LightlyActive
  geom_point(color = "#07F49E") +
  scale_fill_manual(values= "#07F49E") +
  geom_smooth(method = "lm", se=FALSE, color = "blue") +
  labs(
    x = "Total Minutes Asleep",
    y = "Lightly Active Minutes"
  ) +
  theme_minimal()

Fairly_active_plot <- ggplot(ActivityxSleep_NZV_fairly, aes(x = TotalMinutesAsleep, y = FairlyActiveMinu
  geom_point(color = "#247C8E") +
  scale_fill_manual(values = "#247C8E" ) +
  geom_smooth(method = "lm", se=FALSE, color = "blue") +
  labs(
    x = "Total Minutes Asleep",
    y = "Fairly Active Minutes"
  ) +
  theme_minimal()

Very_active_plot <- ggplot(ActivityxSleep_NZV_very, aes(x = TotalMinutesAsleep, y = VeryActiveMinutes))
  geom_point(color = "#10CB98") +
  scale_fill_manual(values = "#10CB98") +
  geom_smooth(method = "lm", se=FALSE, color = "blue") +
  labs(
    x = "Total Minutes Asleep",
```

```
    y = "Very Active Minutes"
  ) +
  theme_minimal()

grid.arrange(Sedentary_plot, Lightly_active_plot, Fairly_active_plot, Very_active_plot, ncol = 2)
```

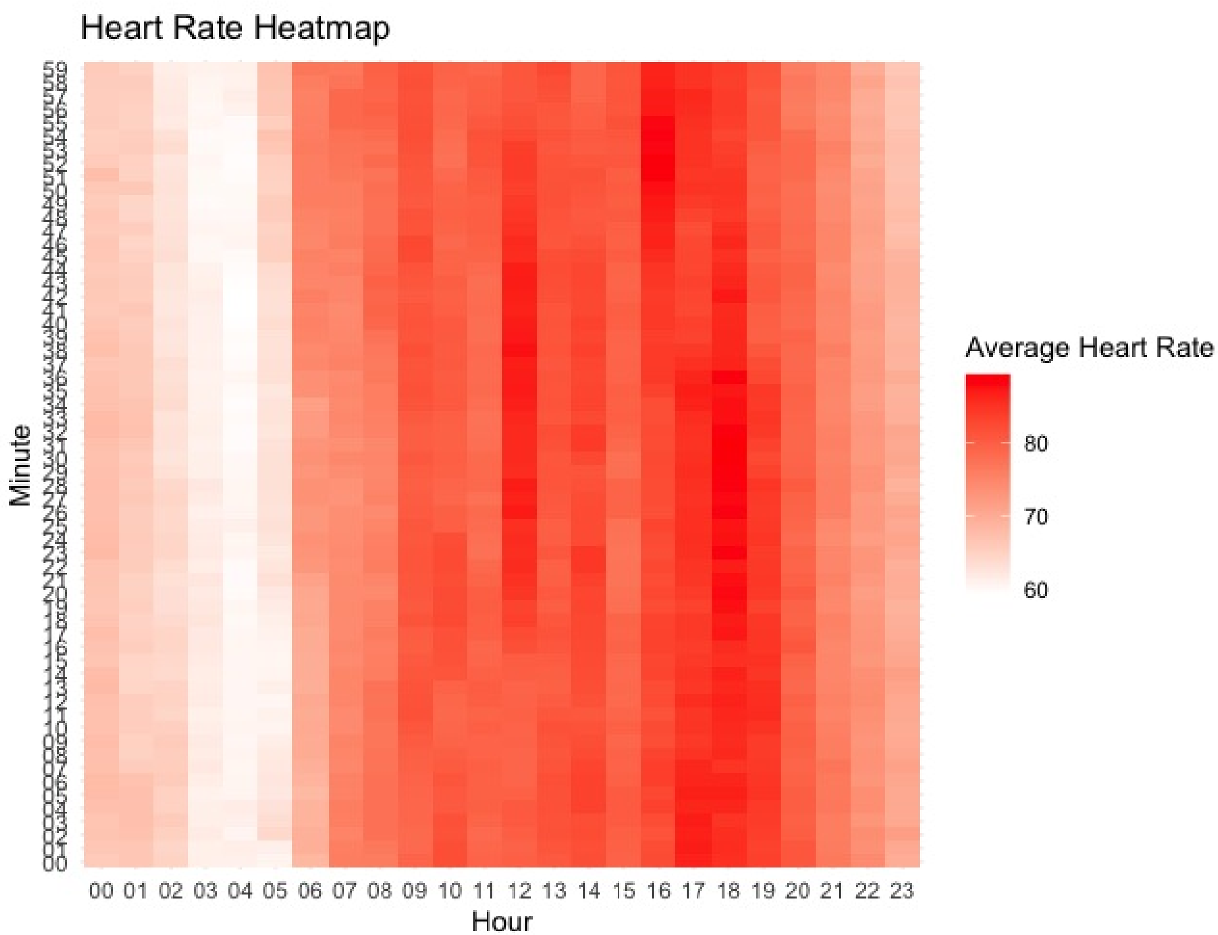


In the first scatterplot we can see that on average the more sedentary the participants were, the less they slept. Whereas, in the other three scatterplots we can't really see a clear influence of activity level on time slept.

Data about Heartrate

```
# Create a heatmap
heatmap_plot <- Heartrate %>%
  mutate(Hour = format(DateTime, "%H"), Minute = format(DateTime, "%M")) %>%
  group_by(Hour, Minute) %>%
  summarize(AverageHeartRate = mean(Value)) %>%
  ggplot(aes(x = Hour, y = Minute, fill = AverageHeartRate)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Heart Rate", x = "Hour", y = "Minute", fill = "Average Heart Rate") +
  theme_minimal()
```

```r
# Print the heatmap
print(heatmap_plot)
```



From the heat map above we can see that the highest heart rate was observed between 17:00 and 19:00 coinciding with activity intensity data which showed that participants tend to be more active between those hours.

Share

Recommendations:

1. Bellabeat can use the information about the users' activity levels and trends during the day/week to send targeted notifications through their app. For example, since we observed that users tend to do more intense physical activity from 17:00 to 19:00 in the day, the app can remind its users a little before 17:00 to do some high intensity physical activity. Furthermore, the app could track the weekly physical activity level of users in order to define their average level of physical activity. Then they could send different notifications/reminders to users of different physical activity levels to make it more personal. Studies Source: Blair et al., 2013 have shown that any level of physical activity can reduce the risk of clinical diseases. While, some activities (and with that intensities) are better for the overall prevention of these diseases, any level of physical activity, be it low or moderate intensity, is better than remaining sedentary. Due to the risk of a sedentary life, it should be one of the priorities to try and minimise it. From the results above, we saw that a big part of the users' day was spent being

sedentary, so it would be a great idea to try and motivate those users to do more low to moderate activities throughout the day.

2. Bellabeat can use the information about users' daily step count to motivate its users to reach their daily goals. I think it would be wise to set a default step goal of 10.000 steps per day, but give users the option to customize it. It could be a good idea, to limit the notifications/reminders about the step count to one per day, as to not overwhelm users.
3. For those users whose goal it is to lose weight, the Bellabeat app could send friendly reminders to engage in high intensity physical activities more often, as we have seen that there is a clear correlation between higher intensity physical activity and burned calories.

Conclusion

It is clear that people use fitness tracking devices for different purposes, so the possibility to customize the goals/aims, daily steps, sleep times, etc. should be a priority and definitely one of the main pillars of the marketing strategy. Furthermore, the Bellabeat app could include a food diary connected to a nutrient database, enabling users to monitor their food consumption and nutrient intake, both very important for a healthy lifestyle.